Title : Rekeying in Secure Mobile Multicast Communications

Description

Field of the invention

This invention relates to rekeying in secure mobile multicast communications and, more specifically to inter-area rekeying of encryption keys.

Background of the invention

The emergence of new Internet applications such as video-conferencing, e-learning and many other applications which are based on group communications experience new challenges such as support of user mobility. A new type of Internet solution is being specified to allow users to communicate with multiple remote hosts while moving in the Internet. In the perspective of deploying multiparty-based applications, the Internet Engineering Task Force (IETF) has defined the IP multicast model [S. Deering, "Host Extension for IP Multicasting", Internet RFC 1112, August 1989]. This model allows any user to send Data Traffic in a single copy of its message to a group of hosts knowing only their group address, or more exactly their multicast address: members join the group by subscription without the sender needing (nor being able) to use the individual group members' unicast addresses. The sender's message is then optimally duplicated by multicast routers on the path towards the receivers.

Unfortunately, the IP Multicast model was originally specified without security support. This issue remains an obstacle for a broader deployment of IP multicast, especially for security-sensitive applications such as Pay-Per-View, private conferences and military communications, for example, where data confidentiality in a dynamic membership context is necessary. Furthermore, the mobility of users complicates the IP multicast security problem.

While general aspects of the multicast security issues [for example T. Hordjono and B. Weis, " The Multicast security (MSEC) Architecture", Internet draft, draft-ietf-msec-arch-04.txt, November 2003] and group encryption key

management issues [for example Mark Baugher and al., "The Group Domain of Interpretation", Internet RFC 3574, July 2003] have been broadly addressed through multiple works and studies, the impact of member's mobility has not been widely considered. The expression 'intra-group mobility' is used herein to refer to movement of member between administrative areas ('inter-autonomous-system mobility'), without leaving or joining a group. The group is normally defined by the multicast address to which the member has subscribed. However, in some circumstances, it would be desirable for a mobile member to remain part of a group defined by the Traffic Encryption Key while changing multicast address.

Such mobility complicates the IP multicast security problem. In fact, inter-autonomous-systemmobility confuses the group membership dynamism since mobile members not only wish to join or leave the multicast group, but may also wish to move within the group between networks or areas while remaining (from a group membership viewpoint) in the secure session. However, member's movement between networks or areas may compromise data secrecy. This would normally require expensive rekeying (the generation of new keying materials) for the existing members in order to ensure data secrecy (Forward and Backward Secrecies). Such a constraint is due to the fact that the underlying group key management service was only designed for stationary members [M. Baugher, R. Canetti, L. Doneti, and F. Lindholm, "Group Key Management Architecture", Internet draft, draft-ietf-msec-gkmarch-06.txt, September 2003].

The Multicast Security (msec) Working Group of the Internet Engineering Task Force (IETF) specifies a common architecture for secure multicast communications [T. Hordjono and B. Weis, "The Multicast security (MSEC) Architecture", Internet draft, draft-ietf-msec-arch-04.txt, November 2003]. This architecture defines a security entity used for delivery and management of cryptographic keys in secure groups. Such an entity is called the Group Controller Key Server (GCKS). The algorithms that manage the distribution, rekeying ('updating'), and revocation of the group keys are known as group key management protocols. The challenge of any key management protocol is to generate and distribute new keys such that the data remains secure while the overall impact on system performance is minimized. There are two basic designs

of group key management model: a centralized design and a distributed design.

In the centralized design, there is a single GCKS that manages the keying material for all the group members.

In a distributed architecture the GCKS entity interacts securely with other GCKS entities to provide more scalable group management services, and hence to avoid signaling overhead and system bottleneck in the case of a large number of group members, which are more likely in the case of a centralized architecture. The manager of the entire group – the domain - is called the Domain GCKS. The domain is divided into administrative regions called areas. The area may be constructed on either a physical or a logical basis. For example, an area may represent a Corporate Network, or may contain a set of users that belong to a common logical group or coalition, independently of their physical location. Each area is managed by an area GCKS, a 'local GCKS'. The present invention relates to the distributed architecture.

In order to ensure the confidentiality of multicast application data, the Domain GCKS generates and distributes to all group members a common key called a Traffic Encryption Key (TEK). This key is used by the multicast source to encrypt data traffic, and by receivers to decrypt source's data. In a distributed scheme, when the Domain GCKS generates the data key (TEK), it multicasts it securely to each area's local Group Controller Key Server (GCKS). The local GCKS is responsible for distributing the TEK to members within its area in a secure fashion using a specific key called: Key Encryption Key (KEK). These keys are used by the local GCKS to encrypt the TEK and subsequent KEK versions (local rekeying). The means by which the local GCKS distributes keys in a secure fashion may include a unicast or multicast secure channel, a logical hierarchy of keys or an extension of the server hierarchy. This framework is depicted in Figure 1 of the accompanying drawings.

The group is dynamic, so that when a new member joins the group, the TEK must be changed to ensure that the newly joining member cannot decrypt previous communications; this requirement is called Backward Secrecy. In the same way, whenever a member leaves the group session, the TEK must be

changed – rekeyed - to ensure that the leaving member cannot decrypt further communications using the TEK it held before leaving the session; such a requirement is called Forward Secrecy. In addition, during a secure multicast session, keys will have a predetermined lifetime and will be periodically refreshed according to particular intervals. This ensures that no keying material remains valid for more than a fixed period of time.

In more detail, when a new member joins the group through a given area it sends the local GCKS a signalling message to request the Group Traffic Encryption Key (TEK) as well as the local KEK. The local GCKS then creates and multicasts a new KEK to all the existing members of its area encrypted with the previous KEK. The new KEK is also unicast to the new member using a secure channel established using suitable mechanisms such as those based on a shared key or member's public key. Once the new KEK is distributed in the relevant area, the Domain GCKS multicasts the new TEK to all local GCKSs and then each local GCKS ($GCKS_i$, $GCKS_j$ and so on) forwards the new TEK to group members in their respective areas in one multicast distribution using the respective KEKs (KEK_i, KEK_j and so on). The process of updating the keying material when a new member joins the group ensures backward secrecy. As a result, the new member cannot have access to an unchanged KEK to decrypt the previous TEK that was encrypted with an unchanged KEK, and thus cannot obtain data transmitted prior to its arrival.

When a member leaves the multicast group, all the valid keys it held just before leaving the session must be changed by notifying its local GCKS. In this case, the member's local GCKS will create and distribute a new KEK to the remaining area members. The proposed solutions for inter-area rekeying we will review here suggest using a unicast secure channel to distribute the new KEK to each remaining member, instead of multicasting, since for both scalability and performance reasons, the GCKS does not have an encryption key shared only with all the remaining members (that is to say all except the leaving member), and hence the GCKS could not use multicast to send the new KEK only to the remaining members. Once the new KEK is distributed, the Domain GCKS generates and distributes a new TEK to all the local GCKSs. Each local GCKS

then forwards securely the new TEK to all its area members using the KEK. This method ensures forward secrecy. That is, a leaving member cannot decrypt future group communications in any area because it does not have the required keying materials (i.e. the new TEK and the new KEK).

In summary, the group key management service is required to ensure the following features in respect of all group members, even for group members that move intra-group between different areas in the domain – inter-area mobility :

- Confidentiality: Only group members can read the multicast traffic.
- Backward secrecy: Every time a new member joins the group, the Traffic Encryption Key (TEK) must be changed for all group members to ensure that the new member cannot decrypt previously transmitted data traffic.
- Forward secrecy: When any group member leaves the multicast group, the TEK must be changed for the remaining group members to ensure that the leaving member cannot decrypt subsequent data traffic.

The situation for intra-group inter-area mobility is illustrated in Figure 2, in which a current group member MM_{ij} initially in the area managed by $GCKS_i$ moves intra-group to the area managed by $GCKS_j$, a current group member MM_{ji} initially in the area managed by $GCKS_j$ moves intra-group to the area managed by $GCKS_i$, and a current group member MM_{kj} initially in the area managed by $GCKS_k$ moves intra-group to the area managed by $GCKS_j$.

It will be appreciated that each area may contain one or more Autonomous Systems (ASs) such as Corporate Networks that may represent a distinct organization with its own physical network. The ASs may also be limited by geographical constraints. Problems have arisen with proposed mechanisms for rekeying, due to security policies, key latency and risks of traffic interruptions and over-frequent inter-area signalling.

One known proposal, referred to as a Static Rekey (SR) approach is illustrated in Figure 3 [C. Zhang and al., "Comparison of Inter-Area Rekeying Algorithms for Secure Wireless Group Communications", IFIP WG 7.3 International Symposium on Computer Performance Modeling, Measurement and Evaluation, September 2000]. In this proposal, the local $GCKS_i$ maintains the KEK_i

unchanged when a mobile member MM_{ij} moves intra-group to a new area. That is, the mobile member MM_{ij} will continue receiving the KEK_i originating from its $GCKS_i$. To achieve this, the $GCKS_i$ may use inter-area multicast to distribute updates of KEK_i for members out of the area. Otherwise, it may use a forwarding node, such as the Home Agent in the case of Mobile IP.

This approach is straightforward and does not require the mobile member MM_{ij} to register with the new local $GCKS_j$ whenever it moves to a new $area_j$. In addition, movement of the member MM_{ij} between the two areas affects neither the members of the previous area nor those of the new one. However, the Static Rekey approach may not work in case where the mobile member moves to a network that belongs to a new administrative area where the security policy restricts the traffic and the interactions with foreign areas. This may be the case when the entire network consists of a coalition composed of loosely connected ASs (e.g., in cooperative military), for example, or when the keys are multicast to the members. Moreover, with the Static Rekey Approach the distribution of keying material to members MM_{ij} out of their areas may suffer latencies or may even fail. Such problems are especially constraining in applications that depend on a rapid dissemination of secure information such as military operations, for example.

In order to reduce both these latencies and inter-AS Signaling, new approaches have been defined proposing mapping the logical area structure directly onto the physical topology of the network. In addition, these approaches propose moving the key changes as close as possible to the members, shifting existing trust relationships to the current location of the mobile member to support future key exchange and eliminating the member's dependence on a distant security server.

If the inter-area movement of the mobile member MM_{ij} were simply considered as a leave from the old area $GCKS_i$ followed by a join to the new area $GCKS_j$, this case would be assimilated to that of a member joining and/or leaving the group altogether. This would introduce an interruption of data transmission as well as additional computations whenever the mobile member transfers between areas. In addition, new TEKs may be generated twice due to the movement of the

member MM_{ij} if the Domain GCKS considers that the entering member is both a member leaving and a member joining the group.

Another known proposal, referred to as an Immediate Rekey (IR) approach is illustrated in Figure 4 [B. DeCleene, L.Dondeti, S. Griffin, T. Hardjono, D. Kiwior, J. Kurose, D. Towsley, S. Vasudevan, and C. Zhang, "Secure Group Communications for Wireless Networks", Proceedings of Military Communications Conference, IEEE MILCOM, Communications for Network-Centric Operations: Creating the Information Force. Vienna, October 2001, pp. 113-117]. The Immediate Rekey algorithm defines new semantics that address member's mobility. In fact, when a group member MM_{ij} moves to a new area, it sends a signaling message to the previous $GCKS_i$ as well as the visited $GCKS_j$. The areas $GCKS_i$ and $GCKS_j$ then update their local KEKs. No TEK rekey is required since the member MM_{ij} proves its group membership to the new local $GCKS_j$ using specific information called credentials, for example in the manner described in S. Griffin, B. DeCleene, L. Dondeti, R. Flynn, D. Kiwior, and A. Olbert," Hierarchical Key Management for Mobile Multicast Members. Accordingly, the data transmission continues uninterrupted.

This approach is simple from a key management point of view. In addition, it separates the case of intra-group mobile members from that of members entering or leaving the group by maintaining the TEK unchanged when a current group member moves between areas. However, it implies systematic rekeying of the local KEKs KEK_i and KEK_j for all group members within the areas affected by the handover of the mobile member MM_{ij}. Thus, the efficiency of this approach is seriously affected in the case of group members with a high inter-area mobility.

Yet another known proposal, referred to as a First Entry Delayed Rekey+ Periodic (FEDRP) approach is illustrated in Figure 5 [C. Zhang and al., "Comparison of Inter-Area Rekeying Algorithms for Secure Wireless Group Communications", IFIP WG 7.3 International Symposium on Computer Performance Modeling, Measurement and Evaluation, September 2000]. The FEDRP approach is like the IR approach in the use of semantics of transfer between areas. It enhances this approach, however, since no local rekeying

occurs in the previous area. That is, when the member MM_{ij} moves intra-group from $area_i$ to $area_j$, it sends one signalling message to $GCKS_i$ and one signalling message to $GCKS_j$. $GCKS_i$ then adds the member MM_{ij} to a list of members that have left the area by intra-group mobility without leaving the group and still hold a valid area key KEK_i for a limited period of time. This list is called an Extra Key Owner List (EKOL). It is reset when and if a member holding a valid area key KEK_i, in $area_i$ or visiting another area, leaves the group and when the timer of the validity period expires. In some cases, the mobile member MM_{ij} may accumulate multiple KEKs that correspond to different areas it has been in. If the mobile member MM_{ij} returns to an area it has previously visited, the local GCKS checks if this member is on the local EKOL list. If this is the case, it will be removed from the list and no rekeying will be needed. In this case, the mobile member MM_{ij} receives (optionally) the current KEK_i using a secure channel. However, if the member MM_{ij} is not present in the list, a new KEK_i is generated and sent in one multicast distribution to current area members using the previous KEK_i, and in a unicast transmission using a secure channel to the mobile member MM_{ij}.

In order to ensure forward secrecy, when a mobile member MM_{ij} visiting another area leaves the group session, all the KEKs it holds must be changed for all the group members holding those KEKs within the affected areas and, in addition, all the EKOL lists holding those KEKs are reset. The KEK_i that the member holds out of $area_i$ may also be changed under specific conditions related to $EKOL_i$ (especially expiration of the key validity period).

The FEDRP approach improves significantly the inter-area rekeying by keeping the KEK_i of the previous area $GCKS_i$ unchanged. However, the FEDRP approach suffers from systematic rekeying when the mobile member enters a new area $GCKS_j$ for the first time. That is, when the member MM_{ij} moves to the new area $GCKS_j$, its mobility is supported in the previous area $GCKS_i$, but not in the visited area $GCKS_j$. In addition, during the session duration, it is more probable that a member enters a given area for the first time than for more than one time. As a result, when a member moves between areas, it is highly probable that a local rekeying occurs in the visited area.

It is desirable to provide a lightweight mechanism that optimizes the computation for mobile members while reducing the impact of their movement on group rekeying

Summary of the invention

The present invention provides a method of and apparatus for inter-area rekeying of encryption keys in secure mobile multicast communications as described in the accompanying claims.

Brief description of the drawings

is a schematic diagram of the architecture of a distributed group key management system, showing Traffic Encryption Key distribution

is a schematic diagram of the group key management system of , showing inter-area movement of group members,

is a schematic diagram of the group key management system of , showing a known rekeying method for inter-area movement of group members,

is a schematic diagram of the group key management system of , showing another known rekeying method for inter-area movement of group members,

is a schematic diagram of the group key management system of , showing yet another known rekeying method for inter-area movement of group members,

is a schematic diagram of a group key management system of the kind shown and for inter-area movement of group members in accordance with one embodiment of the present invention, given by way of example, is a flow chart of an example of a rekeying process when a member joins an area in the system of ,

is a flow chart of an example of a rekeying process when a member leaves an area in the system of , and

is a flow chart of an example of a traffic key reception process in the system of .

Detailed description of the preferred embodiments

Embodiments of the present invention shown in the drawings enable a reduction in the computational capabilities needed for both the key server and area members to support encryption/decryption operations due to membership dynamism (group join/leave) and member's frequent mobility, by separating member's mobility treatment from group membership dynamism, and by amortizing the movement impact over the TEK validity period. In addition embodiments of the present invention provide the following features.

- Reduced impact of member's mobility on group rekeying: the key management system, by separating member's mobility treatment from group membership dynamism (group join/leave), facilitates movement of mobile members between administrative areas while remaining (from a group membership viewpoint) in the group. In addition, when the mobile member leaves the group, the rekeying process reacts with a limited additional impact on the remaining group members. This residual impact is typically due to the accumulated information that the leaving mobile member got from the different areas it visited.
- Reduced computational requirements for mobile members: mobile members have generally very limited resources (e.g. memory space, and computational power). Hence, it is useful to minimize for mobile members both the memory space requirements (e.g. number and size of stored keys) and computations (those involving cryptographic algorithms in particular).
- Trust in Mobile members: the group key management system foresees a level of trust to attribute to mobile members because they may move across different managed areas, and accumulate information about local security services of the different areas they visit. In particular, for some applications, such as military communications, the mobile member must be prevented from continuing to hold valid keying material that it got from a specific security server when the mobile member is out of the management area of that server for more than a given period.

More particularly, embodiments of the present invention provide the following

enhancements:

- avoiding rekeying the members of the visited area each time a mobile member enters it;
- amortizing the impact of member's movement over the TEK lifetime;
- providing a conditional self-generation of local keys for mobile members to reduce signalling between the local GCKS and its members;
- saving resource consumption using derived key-based rekeying for mobile group members.

This mechanism enables the impact of member's movement on area member rekeying and that of the visited area in particular to be reduced significantly. It separates the rekeying of mobile members from membership dynamism (group join/leave) without affecting the KEK rekeying process. This is achieved by introducing a specific key called Visitor Encryption Key (VEK), which is provided to mobile members when they move to a new area.

In more detail, as shown in Figure 6, when the mobile member MM_{ij} moves form $area_i$ to $area_j$, it sends a signalling message to $GCKS_i$ of the area it is leaving as well as a signalling message to $GCKS_j$ of the area in which it is arriving to notify them about its mobility. The signalling messages may be implemented as in FEDRP and IR, using credentials, for example as described in S. Griffin, B. DeCleene, L. Dondeti, R. Flynn, D. Kiwior, and A. Olbert," Hierarchical Key Management for Mobile Multicast Members". Once MM_{ij} is in the new $area_j$, the $GCKS_j$ sends the Visitor Encryption Key VEK_j rather than the KEK_j to the mobile member using a secure channel. This key VEK_j acts similarly to a KEK within this $area_j$ but is not used by members MM_j already in the $area_j$ and possessing the current KEK_j.

There are two types of owner lists for the GCKSs: EKOL (for example similar to that described in B. DeCleene, L.Dondeti, S. Griffin, T. Hardjono, D. Kiwior, J. Kurose, D. Towsley, S. Vasudevan, and C. Zhang, "Secure Group Communications for Wireless Networks", Proceeding of Military Communications Conference, IEEE MILCOM, Communications for Network-Centric Operations: Creating the Information Force. Vienna, October 2001, pp. 1 13-117) and in

addition a Visitor-Key Owner List 'VKOL' that distinguishes between group members MM_i situated in the respective group key management $area_i$ and group members MM_{ij} that were situated in the respective group key management $area_i$ $area_i$ but are visiting another area $area_j$. In this embodiment of the present invention, Visitor-Key Owner Lists (such as $VKOL_i$) contain the list of members still holding a valid VEK (such as VEK_i) but which have subsequently left the key area ($area_i$) in which they obtained the VEK. It is within the scope of the present invention, however, to modify the system to function in other ways, for example so that the Visitor-Key Owner Lists (such as $VKOL_i$) contain the list of members still holding a valid VEK (such as VEK_i) and which are still in the key area ($area_i$) in which they obtained the VEK. This may assume that $GCKS_i$ can distinguish between members holding VEK_i and are out of $area_i$ from those that are in. For example, when the $VKOL_i$ is reset (as described below) only members with VEK_i and that are out of $area_i$ will be removed.

The flow charts shown in Figures 7 to 9 show in more detail examples of the processes that the local GCKS performs in the embodiment of Figure 6 when a new member enters or leaves the area. The processes shown include Group join and leave, as well as movement of a current group member between two areas.

In the embodiment of the invention shown in Figure 6, when a mobile member MM_{ij} moves intra-group from $area_i$ to $area_j$, the following processes will be triggered.:

- if this mobile member MM_{ij} holds a VEK_i, the $GCKS_i$ places the member in the $VKOL_i$ list.
- if the mobile member MM_{ij} holds a KEK_i, the $GCKS_i$ places that member in the $EKOL_i$ list.

In both cases, the $GCKS_i$ triggers a validity period (as in FEDRP) for the local key VEK_i or KEK_i that the mobile member MM_{ij} holds. Subsequently, if the mobile member returns to $area_i$ during the validity period, the $GCKS_i$ removes this member from the list and no local rekeying occurs (optionally, $GCKS_i$ provides the entering member with the current TEK). Accordingly, it is unnecessary to rekey area members when a mobile member returns to a previously visited area.

However, if the mobile member remains out of the $area_i$ and the period expires, the local $GCKS_i$ resets the owner list $EKOL_i$ or $VKOL_i$ and distributes a new local key (KEK_i or VEK_i) to each concerned local member using a secure channel.

Once the mobile member MM_{ij} enters the $area_j$, we may distinguish two cases:

- If there are no VEK_j-members, the $GCKS_j$ generates a VEK_j key (rather than a new KEK_j) and sends it (optionally, along with the current TEK) to the visiting member MM_{ij} in a secure channel, and the KEK_j remains unchanged.
- If VEK_j-members exist, the $GCKS_j$ checks first whether the current VEK_j was used to encrypt the previous TEK. If it is not the case, the $GCKS_j$ will provide the entering mobile member MM_{ij} the current VEK_j. Otherwise, the $GCKS_j$ will provide the entering mobile member MM_{ij} a new VEK_j derived from the current value of VEK_j (preferably using a one-way hash function, for instance: $VEK_{New}=Hash(VEK_{Current})$). During the validity period of the current TEK, the local $GCKS_j$ may not derive more than once a new VEK_j value whatever the number of new mobile members that enter $GCKS_j$'s $area_j$ during the TEK validity period, since the visited $GCKS_j$ is unaware when the visiting mobile member MM_{ij} joined the group in a different area.

Note that in all cases, no local rekeying (KEK_j or VEK_j) is triggered whenever a mobile current group member MM_{ij} moves between two areas. Besides, both Backward and Forward secrecies are ensured. Hence, this embodiment of the present invention separates fully member's intra-group mobility from group membership dynamism (group join/ leave).

When a new member (as opposed to a current group member entering the area by intra-group mobility) joins the group via a given $area_i$ where there are VEK-members, the local $GCKS_i$ will distribute a new KEK_i to all the area members encrypted separately with the current KEK_i and the current VEK_i. As a result, all the current area members will then hold the new KEK_i. The KEK_i is distributed by unicast to the new member as well using a secure channel. Next, the Domain GCKS multicasts securely the new TEK to all the area GCKSs. Each area GCKS

then multicasts the new TEK to all the group members using the local area keys (local KEKs, and local VEKs when and where there are VEK-members).

When any group member leaves the multicast group (as opposed to a current area member leaving the area by intra-group mobility), all the area keys it holds are changed within the affected areas. In this way, for a given $area_i$, the $GCKS_i$ sends either a new KEK_i or a new VEK_i (depending on which local key the leaving member holds) to the concerned members of the $area_i$ using a secure channel with each member, which may be any secure channel except that based on the compromised encryption key (current VEK/KEK), for example a unicast channel. Next, the Domain GCKS multicasts securely the new TEK to all the area GCKSs. Each area GCKS then multicasts the new TEK to all the group members using the local area keys (local KEKs, and local VEKs when and where there are VEK-members) In circumstances where it is desired for a mobile member to be able to change multicast address while keeping the same Traffic Encryption Key TEK, it is unnecessary to send the mobile member a new TEK or to rekey the TEK.

If the rekey is a KEK_i rekey, all the local members will be sent the new KEK_i and the local VKOL list is reset (previous members who visited the $area_i$ are absent will no longer hold a valid VEK_i); in this case, any group members still in the $area_i$ that hold a VEK_i will switch to the new KEK_i. Once in place, the local $GCKS_i$ distributes the new TEK in one multicast transmission using the local keys.

However there is no need to change the KEK_is of the $area_i$ if the leaving member held only the VEK_i. Hence, we save resource consumption since the local KEK_i remains unchanged. In fact, in prior art approaches, when a member leaves the group session, a new local KEK is distributed individually for each remaining area member before a new TEK is multicast to these members. For some applications, the multicast traffic is interrupted until the new TEK will be distributed. This may increase session interruption latency particularly when the number of remaining area members is important. With this embodiment of the present invention, when the member leaving $area_i$ holds a VEK_i and not a KEK_i, the session interruption latency is significantly reduced whether or not the number

of KEK_i-members in the affected $area_i$ is significant. Such a gain of processing time is particularly valuable in real-time applications.

For both group join and leave cases, the $EKOL_i$ as well as the $VKOL_i$ lists on which the leaving member is listed are reset when the new local key is distributed to $area_i$ members.

In summary, in this embodiment of the present invention, the VEK_i can be considered as a temporary key, since the members holding such a key in a given area will switch to the new KEK_i when a KEK_i rekeying occurs (after a group join or periodic rekeying). Any additional processing that management of the VEK_i may introduce in the $GCKS_i$ is negligible.

It is within the scope of the present invention, however, to modify the system to function in other ways with respect to VEK management, for example, so that a new member receives an updated VEK rather than an updated KEK. Another modification would suggest that when a member holding a VEK_i leaves the group, the $GCKS_i$ may distribute a new KEK_i (rather than updating VEK_i as described in our basic mechanism) for all the remaining $area_i$ members using a secure channel with each of those members. Similarly, when a member leaves the group via $area_i$ while holding a KEK_i, $GCKS_i$ may securely unicast a new KEK_i for KEK_i-members only, and VEK_i remains unchanged for VEK_i-members. In this case, $VKOL_i$ list will not be reset. In another option, when a member joins the group via $area_i$, $GCKS_i$ may multicast a new KEK_i encrypted with KEK_i only, and VEK_i remains unchanged for VEK_i-members. In this case, $VKOL_i$ list will not be reset.

Each time the local $GCKS_i$ needs to forward a new TEK (TEK rekey) within its $area_i$ that includes VEK_i-members, it multicasts the new TEK separately encrypted with KEK_i and VEK_i. To achieve this, the local $GCKS_i$ checks first if it has obtained the current VEK_i by derivation from the previous one since the previous TEK forwarding.

If so, the local $GCKS_i$ notifies the VEK_i-members (within the TEK distribution message) that the new TEK they are receiving is encrypted with a new VEK_i (derived from the previous one see above). The VEK_i-members then decrypt this new TEK using the derived VEK_i (the members obtain the new VEK_i by applying

the same function to the previous VEK_i as the one that was applied by the $GCKS_i$ server).

If no derived VEK_i value has been generated since the previous TEK forwarding, it means that the VEK_i has not been changed since then. Thus, the $GCKS_i$ encrypts the TEK_i with the current value of the VEK_i and multicast it to VEK_i-members, notifying them not to obtain a derived VEK_i.